

REMARKS

This paper responds to the Office Action dated March 4, 2004. A diligent effort has been made to respond to each of the rejections contained in the Office Action. It is believed that this Amendment overcomes those rejections and thus places this case in condition for allowance. Reconsideration is respectfully requested.

1. Status of the Claims

Claims 1-43 are cancelled. New claims 44-74 are presented herein.

2. Affidavit of Prior Invention

The previously-submitted Rule 131 Affidavit is hereby withdrawn in view of the Office Action's citation of US 6,473,609 to Schwartz ("Schwartz").

3. Rejections over Lowery and Schwartz

The rejections over Lowery and Schwartz set forth in the Office Action are traversed, particularly in view of the newly presented claims 44-74, which are clearly distinguishable over these two references.

a) Lowery

Lowery (US 6,446,111) describes a method of transmitting executable applets between a client device having minimum memory capabilities and a web server that has been modified to generate applets instead of traditional web page documents, such as HTML documents. Figure 1 of Lowery shows such a configuration in which the client 12 is in direct communication with the modified web server 18. The modified web server 18 can retrieve data from other sources 22, 24, and may use this data in the course of generating the executable applet that is provided to the client.

Lowery, however, does not disclose or suggest the translation of standard page-rendering language documents, such as HTML documents, into a programmatic language, such as JAVA

or some other form of executable program code. Moreover, there is no translation at all in Lowery, because Lowery generates executable code directly, it does not translate fetched HTML or any other type of page-rendered code into a programmatic code. The system described in Lowery could not be utilized with any web site other than one that has been specifically modified to generate an executable applet, whereas the system and method described and claimed in the present application could be used with virtually any web site due to the provision of the translation function which converts the page-rendered code into executable program code.

As described in the presently presented claims 44-74, the plurality of web sites, or other information sources, that may be accessed maintain web page data in standard HTML or other types of standard page-rendering formats and therefore do not require any modification. The gateway/proxy computer, or other part of the system, retrieves the standard HTML-type web pages or other information pages from these plurality of sources, and then translates the page-rendering content into a programmatic language for execution by the client mobile device. In this manner, the client machine can access virtually any web site, regardless of its formatting, because the process of translating the page-rendering code into programmatic code has been centralized at the gateway or other part of the system. Lowery's system by distinction, can only access content that is generated by a modified web server that is able to generate an executable applet according to Lowery's specifications. Therefore, claims 44-74 are distinguishable over Lowery.

b) Schwartz

Schwartz describes a client-server system in which a wireless client does not have a standard web browser but can still access any standard page-rendered web site. In Schwartz, a link server is interposed between the wireless network and the wired network and operates like a

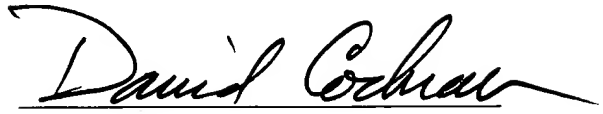
proxy for the wireless client. The wireless client transmits an URL request to the proxy link server, which retrieves the page-rendered code from any web server. This page-rendered code is then compacted into a smaller version of the page-rendered code using one or more converters. The compact page-rendered code is then transmitted to a special interface engine on the client which is capable of rendering a display using the compacted data.

Schwartz, however, does not disclose or suggest the possibility of converting the page-rendered code into programmatic code that can be directly executed by the client device. In Schwartz, the page-rendered code is merely converted into a more compact form of smaller or less page-rendering code segments. The SDD data that is generated by the link server in the Schwartz reference is still page-rendered code, not programmatic code: "Typically, the screen commands are expressed in a form of screen description data (SDD) **that is rendered** in an interface engine in mobile device 350." (Col. 9, ll. 36-40; emphasis supplied) Indeed, column 9, lines 50-58 of Schwartz provide an example "ASCII-like" representation of an SDD file that clearly shows that it is formatted in a page-rendered language, not a programmatic one that is executable. In Schwartz, the link server takes an HTML file and generates a compact data file which is then rendered by the interface engine, it does not convert the page-rendered file into an executable program as set forth in claims 44-74. Therefore, claims 44-74 are distinguishable over Schwartz.

It is believe that this application is in condition for allowance. Should any issues remain, however, the Examiner is invited to phone the undersigned attorney responsible for this application to resolve these issues.

Respectfully submitted,

JONES DAY

A handwritten signature in black ink, reading "David Cochran", written over a horizontal line.

David B. Cochran
(Reg. No. 39,142)

Jones Day
North Point, 901 Lakeside Avenue
Cleveland, Ohio 44114
(216) 586-7506